Astrophysical Gasdynamics Notes:
5. Computational Fluid Dynamics

December 12, 2007

1 Introduction

Since analytical solutions to the fluid equations are only possible in a few limited, idealized cases, the development of computers has led to a whole new branch of fluid studies. In fact, fluid problems were among the first to be tackled with computers (Manhattan project).

Unfortunately, the complexity of the equations also makes them hard to solve numerically. This has led to a large variety of methods for doing computational fluid dynamics (CFD). It can be quite bewildering for a newcomer since the naming of the methods is often obscure and inconsistent.

Here I will look at some of the basic principles which should give you an impression of what CFD is about.

2 Upwind and CFL condition

We start with a simple advection equation in one dimension (1D)

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} = 0 \qquad (1)$$

To solve this numerically we introduce a discretization

$$\begin{aligned} x_j &= x_0 + j\Delta x && (2)\\ t_n &= t_0 + \sum_{n'=0}^{n-1} \Delta t_{n'} && (3) \end{aligned}$$

With this discretization we can write for example

$$\begin{aligned} \frac{\rho_j^{n+1} - \rho_j^n}{\Delta t_n} &= -u\frac{\rho_{j+1}^n - \rho_j^n}{\Delta x} \qquad \text{FTFS} && (4)\\ &= -u\frac{\rho_j^n - \rho_{j-1}^n}{\Delta x} \qquad \text{FTBS} && (5)\\ &= -u\frac{\rho_{j+1}^n - \rho_{j-1}^n}{2\Delta x} \qquad \text{FTCS} && (6) \end{aligned}$$

where FTFS stands for Forward Time, Forward Space, FTBS for Forward Time, Backward Space, and FTCS for Forward Time, Centered Space. If one tries these three approaches one finds that

1. FTFS is always unstable for $u > 0$, and sometimes stable for $u < 0$
2. FTBS is always unstable for $u < 0$, and sometimes stable for $u > 0$
3. FTCS is always unstable
4. FTFS for $u < 0$ and FTBS for $u > 0$ are only stable if $\Delta t_n < \Delta x/|u|$

These results imply that

1. One must take into account the *direction* of the flow (a concept known as 'upwind').
2. One must make sure that within Δt_n one cell only communicates with its neighbours. This is of course related to the *speed* of the flow.

Condition 2 leads to the so-called Courant-Friedrich-Lewy (CFL) condition

$$\Delta t < \frac{\Delta x}{\max(|a|)} \tag{7}$$

where a are all the flow speeds in the problem. One often writes

$$\Delta t = \eta_{\mathrm{CFL}} \frac{\Delta x}{\max(|a|)} \tag{8}$$

with $\eta_{\mathrm{CFL}} < 1$, the so-called CFL number of the calculation.
For the advection equation $a = u$. However for the Euler equations one also needs to take into account the sound speed:

$$a = \max(|u - c_s|, |u + c_s|) \tag{9}$$

This connects to the concept of the domain of dependence: only the region of space-time bounded by the lines $u - c_s$ and $u + c_s$ can be reached by the point under consideration. If $\Delta t > \Delta t_{\mathrm{CFL}}$, point x_j should also affect x_{j+2}, but the methods FTBS and FTFS do not accomodate this. Hence their unstable character in this case.
Figure 1 illustrates something else: for the Euler equations there is not a *single* flow direction. Depending on the values of u and c_s, the flow directions may be all positive (needing FTBS), all negative (needing FTFS), or some positive and some negative. Clearly a rather advanced method that can deal with multiple flow directions is needed for the Euler equations.

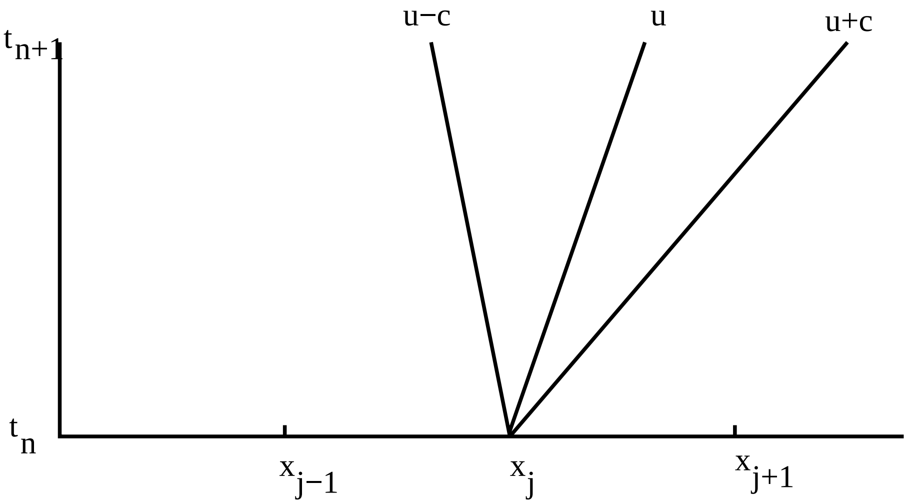

Figure 1: Characteristic directions in space-time delimiting the domain of dependence.

3 von Neumann stability analysis

How can we easily analyze the stability of a given method? This can be done using the von Neumann stability analysis (linear stability analysis). Let's consider our advection equation with the FTFS method (Eq. 4), and write ρ_j^n as $C_k^n \exp(ikj\Delta x)$, i.e. similar to the linear stability analysis used before, but now on a discrete mesh. The idea of the von Neuman method is to evaluate the ratio $R = |C_k^{n+1}|/|C_k^n|$. If R is larger than 1 for all cases, the solution will go to infinity for large n, and is unstable. Substituting into Eq. 4 gives

$$C_k^{n+1} e^{ikj\Delta x} = C_k^n e^{ikj\Delta x} - \lambda \left(C_k^n e^{ik(j+1)\Delta x} - C_k^n e^{ikj\Delta x} \right) \tag{10}$$

where $\lambda = u\Delta t/\Delta x$. So

$$\begin{aligned} C_k^{n+1} &= C_k^n - \lambda \left(C_k^n e^{ik\Delta x} - C_k^n \right) && (11) \\ \frac{C_k^{n+1}}{C_k^n} &= 1 - \lambda(e^{ik\Delta x} - 1) && (12) \\ &= 1 + \lambda - \lambda \cos(k\Delta x) - i\lambda \sin(k\Delta x) && (13) \\ \left| \frac{C_k^{n+1}}{C_k^n} \right|^2 &= (1 + \lambda - \lambda \cos(k\Delta x))^2 + (\lambda \sin(k\Delta x))^2 && (14) \\ &= 1 + 2\lambda(1+\lambda)(1 - \cos(k\Delta x)) && (15) \end{aligned}$$

which is only always smaller than 1 for $2\lambda(1+\lambda) \leq 0$, or

$$-1 \leq \frac{u\Delta t}{\Delta x} \leq 0 \tag{16}$$

meaning that u has to be negative, and the CFL condition needs to hold.

4 Conservation

Let us look at the Euler equations which for a 1D, cartesian coordinate can be written as

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{17}$$

where $\mathbf{W} = (\rho, \rho u, E)^T$ and $\mathbf{F} = (\rho u, \rho u^2 + p, (E+p)u)^T$. A more general case also accomodates for *source terms*

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S} \tag{18}$$

where $\mathbf{S}$ could be external source terms (force, heating, cooling, etc.), or geometric source terms (non-intertial frame of reference, curvilinear coordinates).

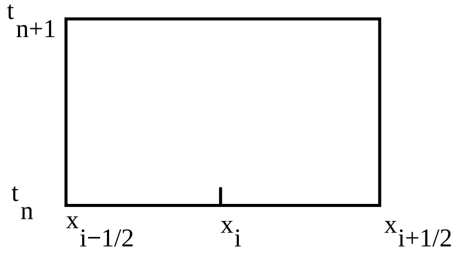

Figure 2: Contour of integration in discretized space-time.

If one defines a cell on our mesh to run from $x_{i-\frac{1}{2}}$ to $x_{i+\frac{1}{2}}$ (with the cell centre at x_i), and a time interval from t_n to t_{n+1}, one can write the integral form of the equations as

$$\int_{t_n}^{t_{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial \mathbf{W}}{\partial t} \mathrm{d}x\mathrm{d}t + \int_{t_n}^{t_{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial \mathbf{F}}{\partial x} \mathrm{d}x\mathrm{d}t = \int_{t_n}^{t_{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{S}\mathrm{d}x\mathrm{d}t \tag{19}$$

where the double integral is actually a closed curve in space-time (Fig. 2).
This gives

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{W}(x, t_{n+1}) - \mathbf{W}(x, t_n)\mathrm{d}x + \int_{t_n}^{t_{n+1}} \mathbf{F}(x_{i+\frac{1}{2}}, t) - \mathbf{F}(x_{i-\frac{1}{2}}, t)\mathrm{d}t = \int_{t_n}^{t_{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{S}\mathrm{d}x\mathrm{d}t \tag{20}$$

Note that we can define the cell averaged state, and the time averaged flux as

$$\langle \mathbf{W}(x_i, t) \rangle = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{W}(x, t)\mathrm{d}x \tag{21}$$

$$\hat{\mathbf{F}}(x, t_{n+\frac{1}{2}}) = \frac{1}{\Delta t} \int_{t_n}^{t_n} \mathbf{F}(x, t)\mathrm{d}t \tag{22}$$

and with these we can write

$$\langle \mathbf{W}(x_i, t_{n+1}) \rangle = \langle \mathbf{W}(x_i, t_n) \rangle + \frac{\Delta t}{\Delta x} \left[\hat{\mathbf{F}}(x_{i-\frac{1}{2}}, t_{n+\frac{1}{2}}) - \hat{\mathbf{F}}(x_{i+\frac{1}{2}}, t_{n+\frac{1}{2}}) \right] + \Delta t \langle \hat{\mathbf{S}}(x_i, t_{n+\frac{1}{2}}) \rangle \qquad (23)$$

which hold *exactly*.

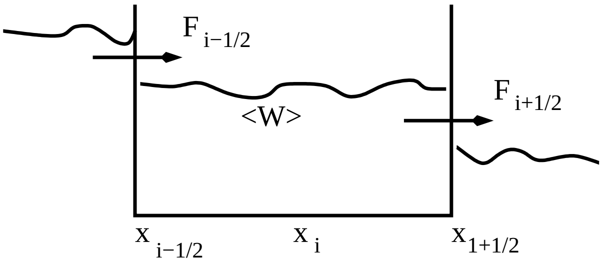

Figure 3: Illustration of the principle of conservative methods.

For $\mathbf{S} = 0$ this form is known as the conservative form, since in it the conserved quantities $\mathbf{W}$ only change because they receive or give to their neighbours. The spatially integrated values remain constant. See Fig. 3 for an illustrative sketch. The effect of the presence of a source term can be illustrated as in Fig. 4.

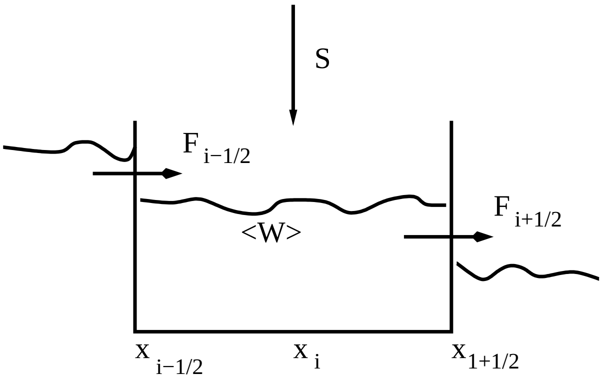

Figure 4: Illustration of the principle of conservative methods with a source term added.

Since the conservative form holds exactly, it is a useful framework for considering and analyzing numerical schemes. If a method is formally dealing with cell-averaged quantities $\langle \mathbf{W}(x_i, t) \rangle$ it is called a *finite-volume* method. If instead it is working with quantities defined at a given position, $\mathbf{W}_i^n = \mathbf{W}(x_i, t_n)$, it is called a *finite-difference* method (see Fig. 5). Note however that at the practical level there is not always a big difference between the two views, or methods may be partly using the finite-volume and partly the finite-difference perspective.

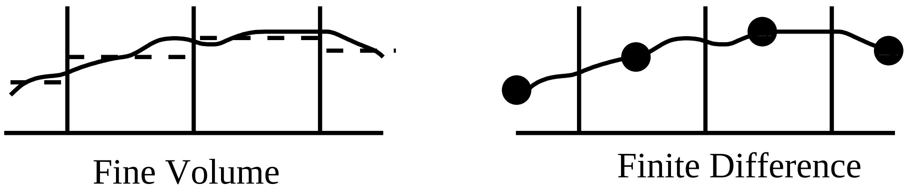

Figure 5: The ideas of finite volume and finite difference.

The conservative form does provide us with a general formula for describing a numerical method

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right) + \Delta t \mathbf{S}_i^n \tag{24}$$

Ignoring $\mathbf{S}$ for now, this formula suggests that our task in designing a numerical hydrodynamics method is about finding a good recipe for $\mathbf{F}_{i-\frac{1}{2}}^{n+\frac{1}{2}}$. Note that this flux in the conservative form is a time-averaged flux

$$\begin{aligned} \hat{\mathbf{F}}(x_{i+\frac{1}{2}}) &= \frac{1}{\Delta t}\int_{t_n}^{t_n} \mathbf{F}(x_{i+\frac{1}{2}}, t)\mathrm{d}t && (25)\\ &\approx \mathbf{F}(x_{i+\frac{1}{2}}, t_{n+\frac{1}{2}}) + O(\Delta t^2) && (26)\\ &\approx \mathbf{F}(x_{i+\frac{1}{2}}, t_n) + O(\Delta t) && (27) \end{aligned}$$

So the more accurate results would be found for a flux value from the intermediate time.

Conservative methods are useful because they automatically conserve the conserved quantities, and the conservative form is also valid at shocks (as can be seen from the shock jump conditions). Since shocks are discontinuities, they are in fact a big challenge for many numerical methods.

5 Lax-Wendroff method

The fact that we need $\mathbf{F}(x_{i+\frac{1}{2}}, t_{n+\frac{1}{2}}) = \mathbf{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ suggests the following simple, two step approximation

$$\text{1.Predictor} \quad \mathbf{W}_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{W}_i^n + \mathbf{W}_{i+1}^n\right) + \frac{\Delta t}{2\Delta x}\left(\mathbf{F}_i^n - \mathbf{F}_{i+1}^n\right) \tag{28}$$

$$\text{2.Corrector} \quad \mathbf{W}_i^{n+\frac{1}{2}} = \mathbf{W}_i^n + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right) \tag{29}$$

where $\mathbf{F}_i^n = \mathbf{F}(\mathbf{W}_i^n)$. This method is known as the Lax-Wendroff method, or sometimes as the Richtmeyer method. It works well for smooth flows, but gives oscillations near shocks and contact discontinuities. To avoid these the concept of *artificial viscosity* was developed. Artificial viscosity for the Lax-Wendroff method consists of replacing step 2 by

$$\mathbf{W}_i^{n+\frac{1}{2}} = \mathbf{W}_i^n + \frac{\Delta t}{\Delta x}\left(\mathbf{F}_{i-\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right) + \epsilon\left(\mathbf{W}_{i+1}^n - 2\mathbf{W}_i^n + \mathbf{W}_{i-1}^n\right) \tag{30}$$

where the last term is the artificial viscosity term. It is called artificial viscosity because its form is identical to that of a real diffusion or viscosity, but its coefficient ϵ is chosen purely for numerical reasons.

Although methods with artificial viscosity gave reasonable results, many felt uncomfortable with the introducing an arbitrary, tunable and unphysical coefficient ϵ. Note also that the above method ignores the lesson we learned from the advection equation: we should take into account the direction of the flow, or rather (in case of the Euler equations), the waves. Method that *do* take this into account are called *upwind methods*.

6 Flux splitting

One approach to implement the upwind idea, is called *flux splitting*. The idea is that the flux vector $\mathbf{F}$ can be written as $\mathbf{F} = \mathbf{F}^+ + \mathbf{F}^-$, where $\mathbf{F}^+$ is the part associated with right-moving waves, and $\mathbf{F}^-$ with left-moving waves (see Fig. 6)

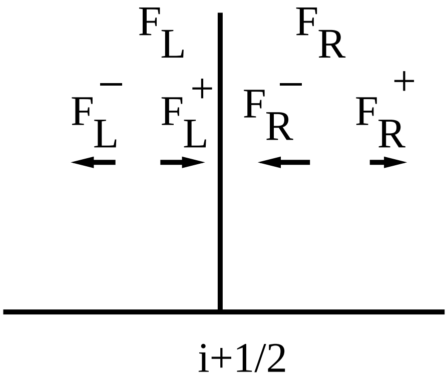

Figure 6: Flux splitting: splitting the fluxes left and right of a cell interface into left and right going fluxes.

Let us consider an interface between two cells. We can define a flux on the left hand side of the interface $\mathbf{F}^L = \mathbf{F}(\mathbf{W}_i)$ (for example), and on the right hand side $\mathbf{F}^R = \mathbf{F}(\mathbf{W}_{i+1})$ (for example). We then split each of these two fluxes into right and left going fluxes: $\mathbf{F}^R = \mathbf{F}^{R+} + \mathbf{F}^{R-}$, $\mathbf{F}^L = \mathbf{F}^{L+} + \mathbf{F}^{L-}$. The only parts that matter are those that cross the interface, so we can take the interface flux to be $\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}^{R-} + \mathbf{F}^{L+}$.

To illustrate this, let us assume that the flow is supersonic and moving to the right on both sides of the interface. Then all waves are right going, so $\mathbf{F}^{R-} = \mathbf{F}^{L-} = 0$, thus the interface flux becomes $\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}^{L+}$ which is correct: no information can travel 'against' a supersonic flow (see Fig. 7). If however the flow is subsonic, some smart expressions for $\mathbf{F}^+$ and $\mathbf{F}^-$ are needed.

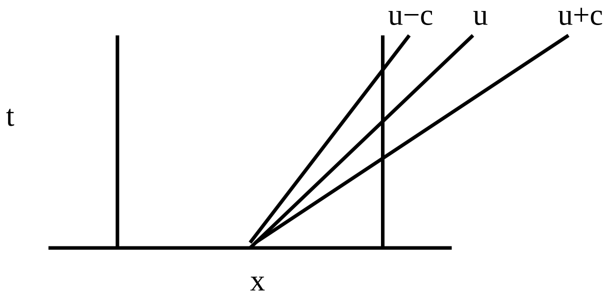

Figure 7: Supersonic flow: all characteristics point to the right. As a consequence no information should be transmitted to the left.

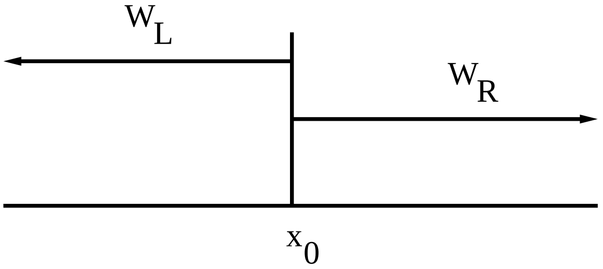

Figure 8: The Riemann problem.

7 Riemann solvers

An even more sophisticated approach is the one used in so-called *Riemann solvers*, or Reconstruction-Evolution methods. This approach, first suggested by Godunov, uses the idea that one can consider the discretized distribution on the mesh as a series of discontinuities. The initial value problem for any discontinuity is known as the Riemann problem,

$$\mathbf{W} = \begin{cases} \mathbf{W}_{\mathrm{L}} & \text{if } x < x_0 \\ \mathbf{W}_{\mathrm{R}} & \text{if } x > x_0 \end{cases} \tag{31}$$

(see Fig. 8) and has known analytical solutions for the Euler equation. If the CFL condition holds, each interface between two cells can be considered to be an isolated Riemann problem (Fig. 9). From the solution of the Riemann problem one finds directly how much mass, momentum and energy flows into a cell from the interface under consideration:

$$\mathbf{F}^{n+\frac{1}{2}}_{i+\frac{1}{2}} = \mathbf{F}(W_{\mathrm{Riemann}}(x_{i+\frac{1}{2}}, t_{n+\frac{1}{2}})) \tag{32}$$

This idea was formulated by Godunov in the 1950s, but was too expensive to implement. In the 1980s computers became fast enough, and also a series of simpler *approximate Riemann solvers* was found. These find an approximate solution to the Riemann problem, good enough to obtain accurate fluxes (which is the only thing needed). The most popular approximate Riemann solver, is the *Roe solver*. A version of the Riemann method popular in astrophysics is known as *PPM* (piecewise parabolic method, Colella & Woodward 1984).

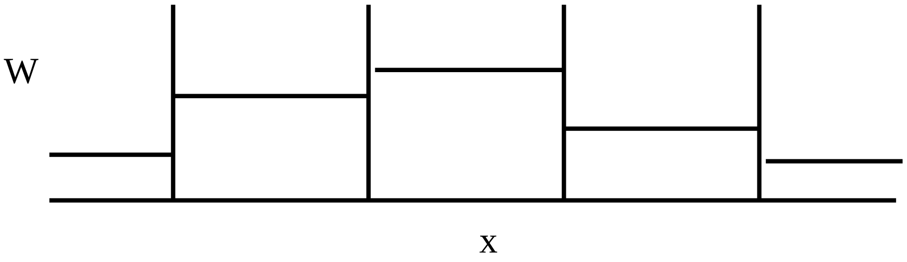

Figure 9: Dividing a mesh into Riemann problems.

8 Multiple dimensions

Up to this point we have only considered 1D methods. How about multi-dimensional methods? In fact, truly multiple dimensional methods are rare. What is most widely used is the technique of operator splitting (or more specifically dimensional splitting). Suppose that the initial value problem to be solved is

$$\frac{\partial f}{\partial t} = \mathcal{L} f \tag{33}$$

with $\mathcal{L}$ some operator that can be written as

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + \cdots + \mathcal{L}_m \tag{34}$$

Then the solution can be found by taking steps

$$\begin{aligned} f^{n+\frac{1}{m}} &= U_1(f^n, \Delta t) && (35)\\ f^{n+\frac{2}{m}} &= U_2(f^{n+\frac{1}{m}}, \Delta t) && (36)\\ f^{n+\frac{3}{m}} &= U_3(f^{n+\frac{2}{m}}, \Delta t) && (37)\\ &\text{etc...} && (38) \end{aligned}$$

where U_k is the solution for $\partial f/\partial t = \mathcal{L}_k f$.
Since the two-dimensional fluid equations can be written as

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \tag{39}$$

we can use the operator splitting approach by first solving for the changes in the x direction, and then using that solution to solve for the changes in the y direction. To avoid systematic effects, the order between these steps can be alternated between time steps. Increased accuracy can be achieved by taking fractional time steps. For two operators:

$$
\begin{aligned}
f^{n+\frac{1}{3}} &= U_1(f^n, \Delta t/2) \quad (40)\\
f^{n+\frac{2}{3}} &= U_2(f^{n+\frac{1}{3}}, \Delta t) \quad (41)\\
f^{n+1} &= U_1(f^{n+\frac{2}{3}}, \Delta t/2) \quad (42)
\end{aligned}
$$

This is sometimes known as *Strang splitting*.

9 Public codes

A number of public hydrodynamics codes exist to allow simulations without the need of writing the code yourself. Well known examples are ZEUS, FLASH, NIRVANA and the PENCIL code. However, the nature of the numerical solvers is still such that one should be cautious when using them as so-called black boxes, especially for types of problems for which they have not been used before.

10 Smooth Particle Hydrodynamics

There exists a technique which does not follow the concepts outlines above. It is called *Smooth Particle Hydrodynamics* and is quite popular in astrophysics (but not in other disciplines). In this technique the fluid is not described on a discrete mesh, but rather as a collection of particles. These particles sample the fluid continuum and each has a density, velocity and pressure attached to it. The particles behave fluid-like because they each influence a region around their current position, described by a kernel function (hence the term *smooth* particles).

One can describe this method as being *adaptive* (particles concentrate in high density areas) and *Lagrangian* (particles follow the flow). It is described more extensively in the Thompson book (copies attached). SPH works well for self-gravitating flows (where particles tend to collapse to form smaller and smaller structures), but has problems with steep gradients such as found in shocks.